



Tilburg University

On the Nucleolus of Neighbour Games

Hamers, H.J.M.; Klijn, F.; Solymosi, T.; Tijs, S.H.; Vermeulen, D.

Publication date:
1999

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Hamers, H. J. M., Klijn, F., Solymosi, T., Tijs, S. H., & Vermeulen, D. (1999). *On the Nucleolus of Neighbour Games*. (CentER Discussion Paper; Vol. 1999-111). Operations research.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

On the Nucleolus of Neighbour Games

Herbert Hamers^a, Flip Klijn^{a,1}, Tamás Solymosi^{b,2},
Stef Tijs^a, and Dries Vermeulen^c

^a*Department of Econometrics and CentER, Tilburg University, The Netherlands*

^b*Department of Operations Research, Budapest University of Economic Sciences, Hungary*

^c*Department of Economics, University of Maastricht, The Netherlands*

Abstract: The class of neighbour games is the intersection of the class of assignment games (cf. Shapley and Shubik (1972)) and the class of component additive games (cf. Curiel *et al.* (1994)). For assignment games and component additive games there exist polynomially bounded algorithms of order p^4 for calculating the nucleolus, where p is the number of players. In this paper we present a polynomially bounded algorithm of order p^2 for calculating the nucleolus of neighbour games.

Keywords: Neighbour Games; Nucleolus

JEL classification: C63, C71, C78

Running title: Nucleolus of Neighbour Games

¹Corresponding author. Department of Econometrics and CentER, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands. Tel. (31) 13 466 2733; Fax. (31) 13 466 3280; e-mail: F.Klijn@kub.nl

²This author's work has been supported by CentER and the Department of Econometrics, Tilburg University and by the Foundation for the Hungarian Higher Education and Research (AMFK).

1 Introduction

This paper provides an $\mathcal{O}(p^2)$ algorithm for calculating the nucleolus of p -person neighbour games. Neighbour games, introduced in Klijn *et al.* (1999), form the intersection of assignment games (cf. Shapley and Shubik (1972)) and component additive games (cf. Curiel *et al.* (1994)). The latter one is the subclass of Γ -component additive games (cf. Potters and Reijnierse (1995)) in which the restricting graph Γ is a line graph. As a consequence, neighbour games have many appealing properties, such as: the core is a non-empty set and coincides with the set of competitive equilibria (Shapley and Shubik (1972)), the core coincides with the bargaining set, and the nucleolus coincides with the kernel (Potters and Reijnierse (1995)). Moreover, neighbour games satisfy the CoMa-property, i.e., the core is the convex hull of all marginal vectors that are in the core (cf. Hamers *et al.* (1999)).

In literature, the computation of the nucleolus of assignment games and component additive games has been discussed extensively. Solymosi and Raghavan (1994) presented an $\mathcal{O}(p^4)$ algorithm for calculating the nucleolus of $p + p$ -person assignment games. An $\mathcal{O}(p^4)$ algorithm for calculating the nucleolus of p -person balanced connected games was provided by Solymosi *et al.* (1998). The class of balanced connected games contains the class of component additive games, and thus the class of neighbour games. This paper provides an $\mathcal{O}(p^2)$ algorithm for calculating the nucleolus of p -person neighbour games. Although the algorithm can be considered as a common specialization of the two algorithms mentioned above, we present it on its own right, since it exhibits special features that neither of the two more general algorithms does. Besides, we give a different line of arguments to see the correctness of the algorithm from those which were used to justify the mentioned more general algorithms.

In Section 2 we provide some preliminaries on cooperative games. Then, in Section 3 we recall the definition of neighbour games and present an $\mathcal{O}(p^2)$ algorithm for finding the nucleolus.

2 Preliminaries

A *cooperative game with transferable utilities* (or game, for short) is a pair (P, v) where $P = \{1, \dots, p\}$ is a finite set of players and $v : 2^P \rightarrow \mathbb{R}$ is a map that assigns to each coalition $S \in 2^P$ a real number $v(S)$, such that $v(\emptyset) = 0$. Here, 2^P is the collection of all subsets (coalitions) of P .

Let (P, v) be a game with a non-empty imputation set $I(v) := \{x \in \mathbb{R}^P : x_i \geq v(i) \text{ for all } i \in P \text{ and } x(P) = v(P)\}$, where $x(P) := \sum_{i \in P} x_i$. For an imputation $x \in I(v)$ and a coalition $S \in 2^P \setminus \{\emptyset\}$ we call $f(S, x) := x(S) - v(S)$ the *satisfaction* of S . Next, let $F(x) := (f(S, x))_{\emptyset \neq S \subseteq P}$ be the vector of satisfactions and let $\theta \circ F(x)$ denote the vector of satisfactions with its elements arranged in non-decreasing order. The *nucleolus* $n(v)$ is then defined by

$$n(v) := \{x \in I(v) : \theta \circ F(x) \succeq_{lex} \theta \circ F(y) \text{ for all } y \in I(v)\},$$

where \succeq_{lex} denotes the lexicographical ordering on $\mathbb{R}^{|2^P|}$. Given two vectors $x, y \in \mathbb{R}^{|2^P|}$ we have $x \succeq_{lex} y$ if either $x = y$ or there exists a k such that $x_i = y_i$ for $i = 1, \dots, k$ and $x_{k+1} > y_{k+1}$.

Schmeidler (1969) proved that if $I(v) \neq \emptyset$, then the nucleolus $n(v)$ is a singleton. Moreover, $n(v)$ is a subset of the core, whenever the latter is not empty. The core of a game (P, v) consists of all vectors that distribute the gains $v(P)$ obtained by P among the players in such a way that no subset of players can be better off by seceding from the rest of the players and act on their own behalf. Formally, the *core* of a game (P, v) is defined by

$$\text{Core}(v) := \{x \in \mathbb{R}^P : f(S, x) \geq 0 \text{ for all } S \subset P \text{ and } f(P, x) = 0\}.$$

A coalition S is called *essential* in the game (P, v) if $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$, and $S_1 \neq \emptyset \neq S_2$ imply that $v(S) > v(S_1) + v(S_2)$. Otherwise, it is called *inessential*. Note that one-player coalitions are always essential. Note also that the inequality related to an inessential coalition in the definition of the core is redundant, i.e., it can be left out without enlarging the solution set of the remaining inequalities. Therefore, for any collection $\mathcal{G} \subseteq 2^P \setminus \{\emptyset\}$ that contains all essential coalitions in the game (P, v) we have

$$\text{Core}(v) = \{x \in \mathbb{R}^P : f(S, x) \geq 0 \text{ for all } S \in \mathcal{G} \text{ and } f(P, x) = 0\}. \quad (1)$$

Huberman (1980) showed that inessential coalitions can also be omitted in the determination of the nucleolus provided the core of the game is not empty. Since in that case the nucleolus lies in the core, the underlying payoff set can be reduced to the core. More precisely,

$$n(v) = \{x \in \text{Core}(v) : \theta \circ G(x) \succeq_{lex} \theta \circ G(y) \text{ for all } y \in \text{Core}(v)\},$$

where $G(x) := (f(S, x))_{S \in \mathcal{G}}$ is the vector of satisfactions of coalitions in \mathcal{G} only.

Let (P, v) be a game with a non-empty core. Let $\mathcal{G} \subseteq 2^P \setminus \{\emptyset\}$ contain all essential coalitions in the game. We need to introduce some notions and notation to be able to describe a simplified version of Kohlberg's (1971) criterion for the nucleolus of such a game. We call a non-empty collection \mathcal{B} of coalitions in P balanced if there are positive numbers $(\lambda_S)_{S \in \mathcal{B}}$ such that $\sum_{S \in \mathcal{B}} \lambda_S e_S = e_P$, where e_S is the vector in \mathbb{R}^P with $(e_S)_i = 1$ if and only if $i \in S$ and 0 otherwise. Given a number $t \geq 0$ and an allocation $x \in \text{Core}(v)$ we define

$$G(x, t) := \{S \in \mathcal{G} : f(S, x) \leq t\}$$

to be the collection of all coalitions in \mathcal{G} whose satisfaction is not more than the given level t at the given core allocation x . In light of Huberman's (1980) simplification it is easy to see that in this setting Kohlberg's (1971) general criterion can be replaced by the following characterization of the nucleolus.

Lemma 2.1 *Let (P, v) be a game with a non-empty core. Let $\mathcal{G} \subseteq 2^P \setminus \{\emptyset\}$ contain all essential coalitions in (P, v) . Then for $x \in \text{Core}(v)$ it holds that $\{x\} = n(v)$ if and only if $G(x, t)$ is balanced for all $t \geq 0$.*

3 Neighbour games and the nucleolus

In this section we provide a polynomially bounded algorithm of order p^2 for finding the nucleolus of neighbour games. We present the algorithm of order p^2 for finding the nucleolus of a special

subclass of neighbour games. After that, we show that we can calculate the nucleolus of an arbitrary neighbour game by breaking up the game in appropriate subgames, applying the algorithm to the subgames, and combining the nucleoli of the subgames. Moreover, we prove that this procedure does not change the computational complexity. Before we present the algorithm we recall the definition and some special features of neighbour games.

Let $P = \{1, \dots, p\}$ be the player set and let $\sigma : P \rightarrow \{1, \dots, p\}$ be an ordering of the players. Obviously, P can be partitioned in the set M of players i in odd position (i.e., $\sigma(i)$ is odd) and the set N of players in even position (i.e., $\sigma(i)$ is even). Players i and j are called *neighbours* if $|\sigma(i) - \sigma(j)| = 1$. We shall use the (unconventional) notation (i, j) if $\sigma(j) = \sigma(i) + 1$, i.e., $(., .)$ is used to indicate the order of (neighbouring) players as given by σ . For all pairs (i, j) let $a_{ij} \geq 0$ be given. Then, a *neighbour game* (P, v) is defined by

$$v(Q) := \max\left\{ \sum_{(i,j) \in \mu} a_{ij} : \mu \in \mathcal{N}(Q) \right\} \quad \text{for all } Q \subseteq P,$$

where $\mathcal{N}(Q)$ is the set of *matchings* of the players in Q in which each matching consists only of pairs (i, j) that are neighbours. From now on the word matching means a matching of this type. A matching $\mu \in \mathcal{N}(Q)$ is called *optimal* for $Q \subseteq P$ if $\sum_{(i,j) \in \mu} a_{ij} = v(Q)$. It is called *minimal* if $a_{ij} > 0$ for all $(i, j) \in \mu$. Note that $v(i) = 0$ for all $i \in P$ and $v(i, j) = a_{ij}$ for all pairs (i, j) .

For the sake of convenience, we assume henceforth that the players in a neighbour game (P, v) are ordered $1 \prec 2 \prec \dots \prec p$.

Example 3.1 Let $P = \{1, 2, 3, 4\}$ be the player set and let σ describe the order $1 \prec 2 \prec 3 \prec 4$. The pairs that are neighbours with respect to σ are $(1,2)$, $(2,3)$, and $(3,4)$. Hence, all other pairs have a worth equal to zero. Take, for instance, $a_{12} = 10$, $a_{23} = 20$, and $a_{34} = 30$. Then the corresponding neighbour game (P, v) is depicted in Table 3.1. The matching that matches player 1 with player 2 and player 3 with player 4 is optimal and minimal.

S	$\{1, 2\}$	$\{1, 2, 3\}$	$\{1, 2, 4\}$	$\{1, 3, 4\}$	$\{2, 3\}$	$\{2, 3, 4\}$	$\{3, 4\}$	$\{1, 2, 3, 4\}$
$v(S)$	10	20	10	30	20	30	30	40

Table 3.1: a neighbour game (P, v) .

◇

Obviously, from the definition of neighbour games it immediately follows that the class of neighbour games is the intersection of the class of assignment games and component additive games. It is also evident that neighbour games are monotonic game (i.e., $v(S) \leq v(T)$ for all $S \subseteq T \subseteq P$) and supperadditive (i.e., $v(S \cup T) \geq v(S) + v(T)$ for all $S, T \subseteq P$ with $S \cap T = \emptyset$).

Since neighbour games are special assignment games, the results of Shapley and Shubik (1972) on the cores of assignment games apply to the cores of neighbour games. In particular, the cores of neighbour games are not empty. Furthermore, they are determined by the inequalities

induced by the one-player and the neighbouring pair coalitions. In other words, for any neighbour game (P, v) the collection

$$\mathcal{G} := \{\{i\} : i \in P\} \cup \{(i, i+1) : i \in P \setminus \{p\}\} \quad (2)$$

contains all essential coalitions of any neighbour game on P . Henceforth, whenever we speak of a coalition it is a singleton or a neighbouring pair of players.

For an optimal matching μ of P we denote, with a slight abuse of notation, by P^+ the set of players that are matched by μ . Define $P^- := P \setminus P^+$, which will be called the set of *isolated* players. The following lemma is a straightforward consequence of a result of Shapley and Shubik (1972).

Lemma 3.2 *Let (P, v) be a neighbour game. Let μ be an optimal matching of P . Let $x \in \mathbb{R}^P$. Then, $x \in \text{Core}(v)$ if and only if the following four conditions are satisfied:*

- (i) $x_i + x_{i+1} = v(i, i+1)$ for all $(i, i+1) \in \mu$;
- (ii) $x_i + x_{i+1} \geq v(i, i+1)$ for all $(i, i+1) \notin \mu$;
- (iii) $x_i = 0$ for all players $i \in P^-$;
- (iv) $x_i \geq 0$ for all players $i \in P^+$.

Next, we present an algorithm for finding the nucleolus for a special class of neighbour games. Let us consider the subclass of neighbour games (P, v) with an even number of players such that the pairs $(1, 2), \dots, (p-1, p)$ form the unique optimal matching for P . Consequently, we must have $v(1, 2), \dots, v(p-1, p) > 0$.

Initial Step of the Algorithm

Input

A neighbour game (P, v) for which p is even and $\mu = \{(2k-1, 2k) : k = 1, \dots, \frac{p}{2}\}$ is the unique optimal matching.

The initial allocation

Compute recursively for $k = 1, \dots, \frac{p}{2}$

$$\begin{aligned}x_{2k-1} &:= \max \{v(2k-2, 2k-1) - x_{2k-2}, 0\} \\x_{2k} &:= v(2k-1, 2k) - x_{2k-1},\end{aligned}$$

where $v(0, 1) := 0 =: x_0$. The vector $x \in \mathbb{R}^P$ is the initial payoff vector.

Satisfaction

Calculate the initial satisfactions of the singletons $f_k := f(\{k\}, x) = x_k$ and the initial satisfactions of the even-odd pairs $f_{2k, 2k+1} := f(\{2k, 2k+1\}, x) = x_{2k} + x_{2k+1} - v(2k, 2k+1)$. Set $\alpha := 0$.

Qualification

Call a coalition

- *settled* if it is an odd-even pair of neighbours,³
- *unsettled* otherwise.

Call an unsettled coalition

- *active* if its satisfaction equals α ;
- *inactive* otherwise.

Before we go on to the inductive step we prove that the initial payoff allocation x is the even friendly core allocation and that the minimum satisfaction level of all unsettled coalitions is $\alpha (= 0)$.

Lemma 3.3 *Let x be the initial payoff allocation of the initial step. Then,*

- (i) *For all $i \in P$, $x_i = v(1, \dots, i) - v(1, \dots, i-1)$.*
- (ii) *$x_i \geq 0$ for every odd player i and $x_i > 0$ for every even player i .*

Proof. First note that in case i is even, we have

$$v(1, \dots, i) = v(1, \dots, i-2) + v(i-1, i), \quad (3)$$

and in case i is odd, we have

$$v(1, \dots, i) = \max \{v(1, \dots, i-1), v(1, \dots, i-2) + v(i-1, i)\}. \quad (4)$$

(i) The proof is by induction on the players. Obviously, the Lemma holds for $i = 1, 2$. Suppose that the Lemma holds for $i = 1, \dots, k-1$, where we may assume that $k \geq 3$. We distinguish

³Henceforth, we say odd-even pair for short. Likewise for even-odd pairs of neighbours.

between two cases.

Suppose i is even. Then,

$$\begin{aligned} x_i &= v(i-1, i) - x_{i-1} \\ &= v(i-1, i) - [v(1, \dots, i-1) - v(1, \dots, i-2)] \\ &= v(1, \dots, i) - v(1, \dots, i-1), \end{aligned}$$

where the first equality follows from the definition of x_k , the second equality from the induction hypothesis, and the third equality from (3).

Suppose i is odd. Then,

$$\begin{aligned} x_i &= \max \{v(i-1, i) - x_{i-1}, 0\} \\ &= \max \{v(i-1, i) - [v(1, \dots, i-1) - v(1, \dots, i-2)], 0\} \\ &= \max \{v(1, \dots, i-2) + v(i-1, i) - v(1, \dots, i-1), 0\} \\ &= \max \{v(1, \dots, i-2) + v(i-1, i), v(1, \dots, i-1)\} - v(1, \dots, i-1) \\ &= v(1, \dots, i) - v(1, \dots, i-1), \end{aligned}$$

where the first equality follows from the definition of x_k , the second equality from the induction hypothesis. The third and fourth equality are obtained by elementary rewriting. Finally, the last equality follows from (4).

(ii) It immediately follows from the definition of x_i that $x_i \geq 0$ for every odd player i . For an even player i it holds that

$$\begin{aligned} x_i &= v(1, \dots, i) - v(1, \dots, i-1) \\ &= v(1, \dots, i-2) + v(i-1, i) - v(1, \dots, i-1) \\ &= v(1, \dots, i-2) + v(i-1, i) - \max \{v(1, \dots, i-2), v(1, \dots, i-3) + v(i-2, i-1)\} \\ &= v(i-1, i) \\ &> 0, \end{aligned}$$

where the first equality follows from the definition of x_k . The second and third equality from (3) and (4), respectively. The fourth equality follows from the fact that μ is the unique optimal matching. Finally, the inequality follows from the observation that $v(1, 2), \dots, v(p-1, p) > 0$.

□

Lemma 3.4 *The initial payoff allocation x is a core allocation. Moreover, it is the even friendly core allocation, i.e., for all $y \in \text{Core}(v)$ and for all even players $i \in N$ it holds that $x_i \geq y_i$.*

Proof. One easily verifies that x satisfies the conditions of Lemma 3.2. This shows that $x \in \text{Core}(v)$.

Suppose there is a core allocation $y \in \text{Core}(v)$ and an even player $i \in N$ such that $x_i < y_i$. Since $(i-1, i) \in \mu$,

$$y_{i-1} = v(i, i-1) - y_i = x_{i-1} - (y_i - x_i) < x_{i-1}, \quad (5)$$

where the equalities follow from Lemma 3.2(i) for $x, y \in \text{Core}(v)$ and the inequality from $x_i < y_i$.

If $x_{i-1} < (y_i - x_i)$, then $y_{i-1} < 0 = v(i-1)$, contradicting $y \in \text{Core}(v)$. So, $x_{i-1} \geq (y_i - x_i) > 0$. Then, by definition of x_{i-1} , we have $x_{i-1} = v(i-1, i-2) - x_{i-2}$. So,

$$x_{i-2} = v(i-1, i-2) - x_{i-1} < v(i-1, i-2) - y_{i-1} \leq y_{i-2},$$

where the first inequality follows from (5) and the second inequality from $y \in \text{Core}(v)$.

So, $x_{i-2} < y_{i-2}$. We can repeat the same argument until we conclude that $y_1 < x_1 = v(1)$, contradicting $y \in \text{Core}(v)$. \square

During the inductive step of the algorithm, which will be spelt out next, we settle singletons and pairs that have not been settled yet. The satisfaction of a coalition that is settled will no longer change during the remainder of the algorithm. The algorithm terminates when all *singletons* are settled.

In every step of the algorithm we deal with a collection of settled and unsettled coalitions. We refine the collection of singletons and pairs that have not been settled yet in two subcollections: a collection of active coalitions and a collection of inactive coalitions. An unsettled singleton or pair is called *active* if it has the minimum satisfaction among all coalitions that are unsettled. Otherwise it is called *inactive*.

We define a *component* to be a maximal set of consecutive players in which each pair of neighbours is settled or active. Note that since odd-even pairs are settled, a component always starts with an odd player and ends with an even player.

The idea of the inductive step is the following. The initial vector is the even friendly core allocation (Lemma 3.4). Hence, in order to obtain the nucleolus we should decrease the satisfaction of the even players and increase the satisfaction of the odd players. For every player i we determine a coefficient d_i that indicates in what direction and with which factor the satisfaction of player i is going to change. After that, a positive number β is determined. The number β depends on the unsettled even players and the inactive (even-odd) pairs. For every player i , the satisfaction is now updated by adding $d_i\beta$ to his current satisfaction. The minimum satisfaction of all unsettled players and pairs is increased with β . Finally, some singletons and pairs might become settled. If this is the case, we verify whether there are still unsettled singletons. If there are no unsettled singletons left, then we are done and the allocation corresponding to the final satisfactions is the nucleolus. Otherwise, we repeat the inductive step.

Inductive Step of the Algorithm: Repeat as long as there is an unsettled singleton:

Input

A qualification of the coalitions in \mathcal{G} as settled, active, or inactive.

All the satisfactions $f_S, S \in \mathcal{G}$.

$\alpha := \min\{f_S : S \text{ unsettled}\}$

Coefficients

Compute for $k = 0, \dots, \frac{p}{2} - 1$,

$$d_{2k+1} := \begin{cases} 0 & \text{if } \{2k+1\} \text{ is settled;} \\ 1 & \text{if } \{2k+1\} \text{ is active and } 2k+1 = 1; \\ 1 & \text{if } \{2k+1\} \text{ is active and } \{2k, 2k+1\} \text{ is inactive;} \\ -d_{2k} + 1 & \text{if } 2k+1 > 1 \text{ and } \{2k, 2k+1\} \text{ is active.} \end{cases} \quad (6)$$

$$d_{2k+2} := -d_{2k+1}. \quad (7)$$

Compute for $k = 1, \dots, \frac{p}{2} - 1$,

$$d_{2k, 2k+1} := d_{2k} + d_{2k+1}. \quad (8)$$

Increase of minimal satisfaction of unsettled coalitions

Compute

$$\beta := \min \left\{ \frac{f_S - \alpha}{1 - d_S} : S \text{ is an inactive coalition and } d_S \leq 0 \right\}. \quad (9)$$

Now, update the satisfactions of all unsettled coalitions:

$$f_S := f_S + d_S \beta \quad \text{for all unsettled } S. \quad (10)$$

Update $\alpha := \alpha + \beta$.

Qualification

If an *unsettled even player* i becomes active, i.e., $f_i = \alpha$, then settle all coalitions from the component's left most (odd) player upto and including player i .

If an *inactive pair* $(i, i+1)$ becomes active and player $i+1$ was already settled, then settle all coalitions from the component's left most (odd) player upto and including coalition $(i, i+1)$.

The next example shows several features of the algorithm.

Example 3.5 Let $P = \{1, \dots, 8\}$ be a player set. In Figure 1, nodes depict the players and the number above an edge denotes the value of the corresponding pair of players. The thick edges correspond with the matched pairs in the optimal matching. The essential information of the neighbour game (P, v) induced by Figure 1 is represented in the first two rows of Table 4.1.

We calculate the nucleolus of (P, v) in Table 4.1. The players and pairs are put in the first row. The values of the pairs are given by the numbers in the second row. Then, we calculate the initial payoff x vector using the initial step. Now we start the inductive step. For an explanation of the concise notation in the table, let us consider rows 4,5, and 6.

In row 4 we depict the satisfactions of all coalitions. If a particular satisfaction is in a box, then the corresponding coalition has already been settled. If a satisfaction has an asterix, then the satisfaction equals α . In row 5 we depict the coefficients, which are determined by (6), (7), and (8). If a satisfaction and the coefficient below are in boldface, then the corresponding coalition determines the number β , using (9). Finally, in row 6 we update the satisfactions using (10). Finally, we settle players using the qualification.

We repeat the inductive step until all singletons are settled. The final payoff vector $(3, 3, 2, 5, 6, 3, 1, 1)$ is the nucleolus of the game (P, v) . Note that coalition $(2, 3)$ has not been settled by the algorithm. \diamond

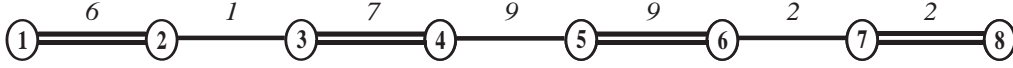


Figure 1: a neighbour game (P, v) .

	1	12	2	23	3	34	4	45	5	56	6	67	7	78	8
v		6		1		7		9		9		2		2	
x	0		6		0		7		2		7		0		2
$\alpha = 0, f$	0*	0	6	5	0*	0	7	0*	2	0	7	5	0*	0	2
$\beta = 1, d$	+1	0	-1	0	+1	0	-1	+1	+2	0	-2	-1	1	0	-1
$\alpha = 1, f$	1*	0	5	5	1*	0	6	1*	4	0	5	4	1*	0	1*
settling													-	-	-
$\alpha = 1, f$	1*	0	5	5	1*	0	6	1*	4	0	5	4	1	0	1
$\beta = 1, d$	+1	0	-1	0	+1	0	-1	+1	+2	0	-2	-2	0	0	0
$\alpha = 2, f$	2*	0	4	5	2*	0	5	2*	6	0	3	2*	1	0	1
settling					-	-	-	-	-	-	-	-			
$\alpha = 2, f$	2*	0	4	5	2	0	5	2	6	0	3	2	1	0	1
$\beta = 1, d$	+1	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0
$\alpha = 3, f$	3*	0	3*	4	2	0	5	2	6	0	3	2	1	0	1
settling	-	-	-												
$\alpha = 3, f$	3	0	3	4	2	0	5	2	6	0	3	2	1	0	1

Table 4.1: calculating the nucleolus of (P, v) .

The next lemma shows, among others, that the inductive step is well-defined and that the algorithm terminates in a finite number of steps ((e), (f), and (g)). The lemma will also be used to prove Theorem 3.8, which states that the resulting payoff vector of the algorithm is the nucleolus.

Lemma 3.6 *In the inductive step of the algorithm:*

- (a) *The coefficients of the odd players in (6) are well-defined, that is:*
 - (i) *player 1 is not inactive;*
 - (ii) *for $k > 1$, player $2k + 1$ and pair $(2k, 2k + 1)$ are not both inactive.*
- (b) *The coefficients defined in (6), (7), and (8) satisfy:*
 - (i) $d_{2k+1} \geq 1$ *for an unsettled odd player $2k + 1$;*
 - (ii) $d_{2k+2} \leq -1$ *for an unsettled even player $2k + 2$;*
 - (iii) $d_{2k,2k+1} = 1$ *for an active pair $(2k, 2k + 1)$;*
 - (iv) $d_{2k,2k+1} \leq 0$ *for an inactive pair $(2k, 2k + 1)$ in the first inductive step;*
 $d_{2k,2k+1} \leq 1$ *for an inactive pair $(2k, 2k + 1)$ in all other inductive steps.*
- (c) $0 < \beta < \infty$.
- (d) *The updated minimum satisfaction level of all unsettled coalitions is $\alpha + \beta$.*
- (e) *At least one unsettled even player gets settled or one inactive pair becomes active.*
- (f) *Active pairs do not become inactive.*
- (g) *Settled coalitions stay settled and the sum of the payoffs of its members does not change.*
- (h1) *If an unsettled even-odd pair gets settled, then it has satisfaction $\alpha + \beta$.*
- (h2) *Suppose that the left most (odd) player $2k + 1$ of a maximal connected set of players that get settled⁴ is unsettled. Then, player $2k + 1$ has satisfaction $\alpha + \beta$.*

Proof. We prove the lemma by induction on the number of steps. First we prove that (a) – (h2) hold for step 1. After that, we assume that (a) – (h2) hold for step $1, \dots, t - 1$ of the algorithm and that there are still unsettled singletons. Then, we will prove that (a) – (h2) also hold for step t of the algorithm.

- (a) (i) Initially player 1 is not settled and $f_1 = x_1 = 0 = \alpha$. Hence, player 1 is active.
- (ii) Suppose that $(2k, 2k + 1)$ is inactive, i.e., $x_{2k} + x_{2k+1} - v(2k, 2k + 1) > 0$. So, $x_{2k+1} > v(2k, 2k + 1) - x_{2k}$. Then, by definition of the initial allocation, $x_{2k+1} = 0$. Hence, $x_{2k+1} = 0 = \alpha$. So, player $2k + 1$ is active.
- (b) (i) Follows from (6) and (7).
- (ii) Since player $2k + 1$ is not settled, we have by (i) that $d_{2k+1} \geq 1$. Then, (7) implies $d_{2k+2} \leq -1$.
- (iii) Since $(2k, 2k + 1)$ is active, we have according to (6) that $d_{2k+1} = -d_{2k} + 1$. By (8) the result follows.
- (iv) Since $(2k, 2k + 1)$ is inactive, we have by (a)(ii) that $2k + 1$ is active. Hence, $d_{2k+1} = 1$ by (7). By (b)(ii) we have $d_{2k} \leq -1$. So, the result follows by (8).

⁴That is, there is no strict superset that is connected and that gets settled as well.

(c) The coalitions that are relevant to calculate β are the even-odd pairs $(2k, 2k + 1)$ that are inactive and the inactive even players, since the odd-even pairs are settled and for the inactive player $2k + 1$ it holds that $d_{2k+1} \geq 1 > 0$ by (b)(i).

Let $(2k, 2k + 1)$ be an inactive pair. Then, since initially $\alpha = 0$, we have $d_{2k,2k+1} \leq 0$ by (b)(iv). Since $(2k, 2k + 1)$ is inactive, $f_{2k,2k+1} - \alpha = f_{2k,2k+1} > 0$. So, $\frac{f_{2k,2k+1}-\alpha}{1-d_{2k,2k+1}} \in (0, \infty)$.

For an inactive even player $2k$ we have $d_{2k} \leq -1$ (by (b)(ii)) and $f_{2k} - \alpha = f_{2k} = x_{2k} > 0$ (by Lemma 3.3). So, $\frac{f_{2k}-\alpha}{1-d_{2k}} \in (0, \infty)$.

Now note that there is at least one unsettled even player and (c) follows.

(d), (e) We write f'_S for the satisfaction of coalition S after the update.

From (c) it follows that there is an active coalition S that is either an inactive pair $(2k, 2k + 1)$ or an even player $2k$ with $\beta = \frac{f_S - \alpha}{1 - d_S} = \frac{f_S}{1 - d_S}$, since $\alpha = 0$. Then, $f'_S = f_S + d_S \beta = f_S + d_S \frac{f_S}{1 - d_S} = \beta = \alpha + \beta$.

So, it is sufficient to prove that for all unsettled coalitions S we have $f'_S \geq \alpha + \beta$.

Let $2k + 1$ be an (unsettled) odd player. By (b)(i), $d_{2k+1} \geq 1$. So, $f'_{2k+1} = f_{2k+1} + d_{2k+1} \beta \geq f_{2k+1} + \beta \geq \alpha + \beta$, where the last inequality follows since player $2k + 1$ is unsettled.

Let $2k$ be an (unsettled) even player. By (b)(ii), $d_{2k} \leq -1$. By definition of β , $\beta \leq \frac{f_{2k}-\alpha}{1-d_{2k}}$. So, $f'_{2k} = f_{2k} + d_{2k} \beta \geq \alpha + \beta$.

Suppose $(2k, 2k + 1)$ is an active pair. By (b)(iii), $d_{2k,2k+1} = 1$. So, $f'_{2k,2k+1} = f_{2k,2k+1} + d_{2k,2k+1} \beta = f_{2k,2k+1} + \beta = \alpha + \beta$, where the last equality follows since $(2k, 2k + 1)$ is active.

Suppose $(2k, 2k + 1)$ is an inactive pair. By (b)(iv), $d_{2k,2k+1} \leq 0$. Then from $\beta \leq \frac{f_{2k,2k+1}-\alpha}{1-d_{2k,2k+1}}$ it follows that $f'_{2k,2k+1} = f_{2k,2k+1} + d_{2k,2k+1} \beta \geq \alpha + \beta$.

(f) follows from (b)(iii) and (d).

(g) By the settling procedure in the initial step, the only settled coalitions are the odd-even pairs. By definition of the settling procedure they stay settled. It follows from (7) that the sum of the payoffs of the players in an odd-even pair does not change.

(h1) Suppose an (unsettled) even-odd pair $(2k, 2k + 1)$ gets settled. Then, by definition of the settling procedure, $(2k, 2k + 1)$ is active. From (b)(iii) it then follows that the satisfaction of $(2k, 2k + 1)$ is equal to $\alpha + \beta$.

(h2) Let $2k + 1$ be the left most (odd) player of a maximal connected set of players that get settled. Then, by the initial settling procedure, player $2k + 1$ is unsettled. If $2k + 1 = 1$, then the satisfaction of player $2k + 1$ is equal to $\alpha + \beta$ by (6). If $2k + 1 \neq 1$, then, by definition of the settling procedure, $(2k, 2k + 1)$ is not active. So, by (a)(ii), player $2k + 1$ is active. Hence, by (d), the satisfaction of player $2k + 1$ is equal to $\alpha + \beta$.

Now assume that (a) – (h) hold for steps $1, \dots, t - 1$ of the algorithm and that there are still unsettled singletons. We prove that (a) – (h) also hold for step t of the algorithm.

(a) As for (i), suppose that player 1 is not settled in step t . Then it follows from the Induction Hypothesis (IH, for short) ((g) and (a)(i)) that player 1 was active in step $t - 1$. Then, by (6), in step $t - 1$ we had $d_1 = 1$. So, player 1 got the minimal increase of satisfactions of unsettled coalitions in step $t - 1$. So, by (d) for step $t - 1$, player 1 is also active in step t .

As for (ii), suppose that in step t player $2k + 1$ is unsettled and $(2k, 2k + 1)$ is inactive. Then it follows from IH ((f) and (g)) that $(2k, 2k + 1)$ was inactive in step $t - 1$. From IH ((g) and (a)(ii)) it follows that player $2k + 1$ was active in step $t - 1$. So, player $2k + 1$ got the minimal increase of satisfactions of unsettled coalitions in step $t - 1$. So, by (d) for step $t - 1$, player $2k + 1$ is also active in step t .

(b) (i) follows from (6) and (7). (ii) follows from (6), (7), and the fact that if player $2k$ is not settled, then player $2k - 1$ is also not settled. (iii) follows from (6) and (8). As for (iv), note that $d_{2k+1} = 1$ by (6). Furthermore, $d_{2k} \leq 0$ by (6) and (7). So, $d_{2k,2k+1} = d_{2k} + d_{2k+1} \leq 0 + 1$ by (8).

(c) Suppose $(i, i + 1)$ is an inactive pair. Then, $(i, i + 1) = (2k, 2k + 1)$ for some k (by IH (g) for steps $1, \dots, t - 1$). Then, $d_{2k,2k+1} \leq 1$ by (b)(iv). Since $(2k, 2k + 1)$ is inactive, $f_{2k,2k+1} - \alpha > 0$. Since we compute $\frac{f_{2k,2k+1} - \alpha}{1 - d_{2k,2k+1}}$ only if $d_{2k,2k+1} < 1$, this ratio is positive and finite.

For an unsettled even player $2k$ we have $d_{2k} \leq -1$ (by (b)(ii)) and $f_{2k} - \alpha > 0$ (by IH (d) for steps $1, \dots, t - 1$ and the definition of the settling procedure). So, $\frac{f_{2k} - \alpha}{1 - d_{2k}} \in (0, \infty)$.

By assumption there are still unsettled singletons. Then, it follows from the definition of the settling procedure and IH (g) for steps $1, \dots, t - 1$ that there is at least one unsettled even player. This proves (c).

(d), (e) The proof is almost a copy of the proof of (d), (e) for step 1, except for the part in which we prove that for every inactive pair $(2k, 2k + 1)$ it holds that $f'_{2k,2k+1} \geq \alpha + \beta$.

Take an inactive pair $(2k, 2k + 1)$. By (b)(iv), $d_{2k,2k+1} \leq 1$. If $d_{2k,2k+1} = 1$, then $f'_{2k,2k+1} = f_{2k,2k+1} + d_{2k,2k+1}\beta = f_{2k,2k+1} + \beta > \alpha + \beta$, where the inequality follows from the fact that $(2k, 2k + 1)$ is inactive and IH (d) for step $t - 1$.

If $d_{2k,2k+1} < 1$, then from $\beta \leq \frac{f_{2k,2k+1} - \alpha}{1 - d_{2k,2k+1}}$ it follows that $f'_{2k,2k+1} = f_{2k,2k+1} + d_{2k,2k+1}\beta \geq \alpha + \beta$.

(f) follows from (b)(iii) and (d).

(g) By definition of the settling procedure the settled coalitions stay settled.

Suppose an odd player $2k + 1$ is settled. By (6), $d_{2k+1} = 0$. So, the payoff of player $2k + 1$ does not change.

Suppose an even player $2k$ is settled. By definition of the settling procedure, player $2k - 1$ is also settled. Then, it follows from (6) and (7) that $d_{2k} = 0$. So, the payoff of player $2k$ does not change.

Suppose an even-odd pair $(2k, 2k + 1)$ is settled. By definition of the settling procedure, players $2k$ and $2k + 1$ are also settled. Then, it follows from the above that the payoffs of players $2k$ and $2k + 1$ do not change. So, the sum of the payoffs of players $2k$ and $2k + 1$ does not change either.

Finally, let $(2k - 1, 2k)$ be a (settled) odd-even pair. By (6) and (7), the sum of the payoffs of players $2k - 1$ and $2k$ does not change.

(h1) Suppose that an unsettled even-odd pair $(2k, 2k + 1)$ gets settled. By definition of the settling procedure, $(2k, 2k + 1)$ is active. From (b)(iii) it then follows that the satisfaction of $(2k, 2k + 1)$ is equal to $\alpha + \beta$.

(h2) Let $2k + 1$ be the left most (odd) player of a maximal connected set of players that get settled. Suppose that player $2k + 1$ is unsettled. If $2k + 1 = 1$, then, by (a)(i), player $2k + 1$ is active. By (6) and (d) it then follows that the satisfaction of player $2k + 1$ is equal to $\alpha + \beta$. If $2k + 1 \neq 1$, then, by definition of the settling procedure, $(2k, 2k + 1)$ is not active. So, by (a)(ii), player $2k + 1$ is either settled or active. Since we have assumed that player $2k + 1$ is unsettled, it follows that player $2k + 1$ is active. Hence, by (d), the satisfaction of player $2k + 1$ is equal to $\alpha + \beta$. \square

The following lemma will be used to show that the outcome of the algorithm is the nucleolus.

Lemma 3.7 *Let (P, v) be a neighbour game. Let $\mathcal{B} \subseteq \mathcal{G}$ be a non-empty collection of coalitions. Then, \mathcal{B} is balanced if and only if for every $T \in \mathcal{B}$ there is a partition \mathcal{C} of P such that $T \in \mathcal{C} \subseteq \mathcal{B}$.*

Proof. First we prove the ‘if’-part. For each $T \in \mathcal{B}$, let \mathcal{C}_T be a partition of P such that $T \in \mathcal{C}_T \subseteq \mathcal{B}$. Let us count how many times a coalition $U \in \mathcal{B}$ appears in the partitions \mathcal{C}_T , $T \in \mathcal{B}$, and let u denote this number. Clearly, $1 \leq u \leq |\mathcal{B}|$. Now it is straightforward to check that the weights $\lambda_U := \frac{u}{|\mathcal{B}|}$ balance the collection \mathcal{B} .

To prove the ‘only if’-part, let \mathcal{B} be balanced. Take any $T \in \mathcal{B}$. Let i denote the right most player in T , i.e., $T = \{i\}$ or $(i - 1, i)$. If $i < p$, then there must be a coalition $U \in \mathcal{B}$ with left most player $i + 1$, since the weight of coalition $(i, i + 1)$ - if it is in \mathcal{B} at all - is strictly less than 1, so player $i + 1$ must also be covered by coalitions in \mathcal{B} disjoint from T . Let j denote the right most player in U , i.e., $j = i + 1$ or $i + 2$. If $j < p$, then we repeat the argument. Eventually we select disjoint coalitions from \mathcal{B} that cover all the players from $i + 1$ upto and including p . A similar argument to the left gives that there are disjoint coalitions in \mathcal{B} that cover all the players on the left of T . We conclude that the condition in the lemma is indeed satisfied. \square

Theorem 3.8 *The final allocation of the algorithm is the nucleolus.*

Proof. Let (P, v) be a neighbour game for which p is even and $\mu = \{(2k - 1, 2k) : k = 1, \dots, \frac{p}{2}\}$ is the unique optimal matching. Let z be the final allocation of the algorithm.

We prove that $n(v) = \{z\}$ by using Lemma 2.1, i.e., we check the balancedness of $G(z, t)$ for all $t \geq 0$. Actually, we verify the balancedness of $G(z, f(S, z))$ for all $S \in \mathcal{G}$. Take $S \in \mathcal{G}$. We have to show that $G(z, f(S, z))$ is balanced. By Lemma 3.7 it is sufficient to show that for every $T \in G(z, f(S, z))$ there is a partition \mathcal{C} of P such that $T \in \mathcal{C} \subseteq G(z, f(S, z))$. So, take $T \in G(z, f(S, z))$. We distinguish between the case in which T gets settled and the case in which it does not get settled in the algorithm.

CASE 1: coalition T gets settled in the algorithm.

Suppose T gets settled in the initial step. Then T is an odd-even pair. Since also all other odd-even pair get settled in the initial step, the collection of these pairs is a partition with the desired property for T .

Now we suppose that T gets settled during the inductive step of the algorithm. We distinguish between the two events in the inductive step that cause T to get settled.

SUBCASE 1.A: an unsettled even player j becomes active.

Let $f'_j = \alpha + \beta > 0$ be his satisfaction. We settle all coalitions from the component's left most (odd) player i upto and including player j . Note that T is one of these coalitions and $f'_T \geq \alpha + \beta$ (Lemma 3.6 (d)).

By Lemma 3.6 (h1), (h2), and (g), the coalitions $\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j-2, j-1\}, \{j\}$ have satisfactions $\leq \alpha + \beta$. This collection together with some odd-even pairs forms a partition, and hence a partition with the desired property for T if $T \in \{\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j-2, j-1\}, \{j\}\}$.

If $T \notin \{\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j-2, j-1\}, \{j\}\}$, then T is some singleton $\{k\} \subseteq \{i, \dots, j\}$, since the satisfaction of odd-even pairs equals 0 and $f'_S \geq \alpha + \beta > 0$. Now $T = \{k\}$ forms together with some odd-even pairs and $\{k+1, k+2\}, \{k+3, k+4\}, \dots, \{j-2, j-1\}, \{j\}$ a partition with the desired property for T if k is odd. If k is even, then $T = \{k\}$ together with some odd-even pairs and $\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{k-2, k-1\}$ is a partition with the desired property for T .

SUBCASE 1.B: an inactive pair $(j, j+1)$ becomes active and coalition $\{j+1\}$ is already settled.

Let $f'_{(j, j+1)} = \alpha + \beta$ be the satisfaction of the pair $(j, j+1)$. We settle all coalitions from the component's left most (odd) player i upto and including coalition $\{j, j+1\}$. Note that T is one of these coalitions and $f'_T \geq \alpha + \beta$ (Lemma 3.6 (d)).

By Lemma 3.6 (h1), (h2), and (g), the coalitions $\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j, j+1\}$ have satisfactions $\leq \alpha + \beta$. Since $\{j+1\}$ is settled, there exists some even player l for which the coalitions $\{j+1\}, \{j+2, j+3\}, \{j+4, j+5\}, \dots, \{l-2, l-1\}, \{l\}$ are settled and have all the same (fixed) satisfaction $z_{j+1} < f'_{(j, j+1)}$. The collection that consists of the coalitions $\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j, j+1\}, \{j+2, j+3\}, \{j+4, j+5\}, \dots, \{l-2, l-1\}, \{l\}$ together with some odd-even pairs forms a partition and hence a partition with the desired property for T if $T \in \{\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j, j+1\}\}$.

If $T \notin \{\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{j, j+1\}\}$, then T is some singleton $\{k\} \subseteq \{i, \dots, j\}$, since the satisfaction of odd-even pairs equals 0 and $f'_T \geq \alpha + \beta > 0$. Now $T = \{k\}$

forms together with some odd-even pairs and $\{k+1, k+2\}, \{k+3, k+4\}, \dots, \{l-2, l-1\}, \{l\}$ a property with the desired property for T if k is odd. If k is even, then $T = \{k\}$ together with some odd-even pairs and $\{i\}, \{i+1, i+2\}, \{i+3, i+4\}, \dots, \{k-2, k-1\}$ is a partition with the desired property for T .

CASE 2: coalition T does not get settled in the algorithm.

Since all singletons and all odd-even pairs get settled, T is an even-odd pair. So, $T = (2k, 2k+1)$ for some k . Note that $(2k, 2k+1)$ is inactive (otherwise we would have settled $(2k, 2k+1)$ in the last step of the algorithm). Since the coalitions $\{2k\}$ and $\{2k+1\}$ are already settled, it follows from the definition of the settling procedure and Lemma 3.6 (h2) that the satisfaction of $(2k, 2k+1)$ is greater than the satisfaction of both $\{2k\}$ and $\{2k+1\}$.

Then, by combining the partitions for $\{2k\}$ and $\{2k+1\}$ appropriately, it readily follows that there is also a partition with the desired property for $(2k, 2k+1)$. \square

Proposition 3.9 *Let (P, v) be a neighbour game for which p is even and $\mu = \{(2k-1, 2k) : k = 1, \dots, \frac{p}{2}\}$ is the unique optimal matching. Then, the algorithm determines the nucleolus of (P, v) in $\mathcal{O}(p^2)$ time.*

Proof. Initially, there are at most $p-1$ inactive singletons and pairs ($\frac{p}{2}$ even players and $\frac{p}{2}-1$ even-odd pairs). By Lemma 3.6 (e) and (f), in every inductive step at least one unsettled even player gets settled or one inactive pair becomes active (and does not become inactive anymore). So, the algorithm terminates after at most $p-1$ inductive steps.

Since both the initial and the inductive step take $\mathcal{O}(p)$ time, the algorithm determines the nucleolus in $\mathcal{O}(p^2)$ time. \square

Now we show that we can calculate the nucleolus of an arbitrary neighbour game by breaking up the game in appropriate subgames, applying the algorithm to the subgames, and constructing the nucleolus out of the nucleoli of the subgames. We prove that this procedure does not change the computational complexity.

For this, let (P, v) be a neighbour game. Let us first consider the relation between the nucleolus and the kernel (Davis and Maschler (1965)) of (P, v) . Since the class of neighbour games is a subclass of the class of component additive games it immediately follows from Potters and Reijnders (1995) that the kernel $k(v)$ coincides with the nucleolus $n(v)$. Using Potters and Reijnders (1995), it is not difficult to prove that

$$n(v) = \{x \in \text{Core}(v) : s_{ii+1}(x) = s_{i+1i}(x) \text{ for all } i = 1, \dots, p-1\}, \quad (11)$$

where $s_{ii+1}(x) = \min \{x(S) - v(S) : i \in S \subseteq P \setminus \{i+1\}, S \text{ connected}\}$ (and $s_{i+1i}(x)$ defined similarly). Note that $s_{ii+1}(x), s_{i+1i}(x) \geq 0$ by the fact that $x \in \text{Core}(v)$. Let μ be an optimal

matching of P . To simplify expression (11), we make the following two remarks for a core allocation $x \in \text{Core}(v)$. First, from Lemma 3.2(iii) it follows that $s_{ii+1}(x) = 0$ for $i \in P^-$, $i \neq p$. For such i , we have that $i + 1 \in P^-$ or $(i + 1, i + 2) \in \mu$. One can verify that in both cases $s_{i+1i}(x) = 0$. Second, if $\{i, i + 1\} \subseteq P^+$, but $(i, i + 1) \notin \mu$, then $s_{ii+1}(x) = 0 = s_{i+1i}(x)$ by using Lemma 3.2(i) and taking $S = \{i - 1, i\}$ and $S = \{i + 1, i + 2\}$, respectively. From these two remarks it follows that (11) can be reduced further to

$$n(v) = \{x \in \text{Core}(v) : s_{ii+1}(x) = s_{i+1i}(x) \text{ for all } (i, i + 1) \in \mu\}. \quad (12)$$

Note, however, that (12) does not directly help us a great deal in calculating the nucleolus. This is because the equations $s_{ii+1}(x) = s_{i+1i}(x)$ contain a lot of cumbersome minimization operations, already for a small number of players.

Nevertheless, expression (12) together with the next lemma shows that in order to calculate the nucleolus of a neighbour game, it suffices to calculate the nucleolus for the subgames with possible isolated players on the extremes and no isolated players in the middle.

Lemma 3.10 *Let (P, v) be a neighbour game. Let $x \in \text{Core}(v)$. Let $i \in P^-$ and $k \in \{i, \dots, p\}$. Then, $s_{kk+1}(x) = \min \{x(j, \dots, k) - v(j, \dots, k) : i \leq j \leq k\}$.*

Proof. Take some connected set $S \supseteq \{i, \dots, k\} =: S'$ with $k + 1 \notin S$. Let μ_S be an optimal matching for S . Define a matching $\mu_{S'}$ for S' by $(j, j + 1) \in \mu_{S'}$ iff $(j, j + 1) \in \mu_S$ and $j \geq i$. Then,

$$\begin{aligned} & (x(S) - v(S)) - (x(S') - v(S')) \\ & \geq \left(\sum_{j \in S} x_j - \sum_{(j, j+1) \in \mu_S} v(j, j+1) \right) - \left(\sum_{j \in S'} x_j - \sum_{(j, j+1) \in \mu_{S'}} v(j, j+1) \right) \\ & = \sum_{j \in S \setminus S'} x_j - \sum_{(j, j+1) \in \mu_S, j < i} v(j, j+1) \\ & \geq \sum_{j \in S \setminus S'} x_j - v(S \setminus S') \\ & \geq 0. \end{aligned}$$

The first inequality follows from the fact that μ_S is an optimal matching for S and $\mu_{S'}$ is a matching for S' . The first equality follows from the definition of $\mu_{S'}$. The second inequality from the fact that $\{(j, j + 1) : (j, j + 1) \in \mu_S, j < i\}$ defines a (possibly non-optimal) matching for $S \setminus S'$. The third inequality follows from the fact that $x \in \text{Core}(v)$. \square

Corollary 3.11 *Let (P, v) be a neighbour game. Suppose there is a player $i \in P^-$ with $1 < i < p$. Let $S^1 := \{1, \dots, i\}$ and $S^2 := \{i, \dots, p\}$. Then,*

$$n(v) = (n_1(v|_{S^1}), \dots, n_{i-1}(v|_{S^1}), 0, n_{i+1}(v|_{S^2}), \dots, n_p(v|_{S^2})).$$

Now let us consider neighbour games with possible isolated players on the extremes and no isolated players in the middle. In the next lemmas we show that we can make a further reduction by proving that it suffices to calculate the nucleolus of a neighbour game that slightly differs from the original game and in which we leave out the isolated players of the original game. In Lemma 3.12 we consider the case in which only the first player is isolated. In Lemma 3.13 we consider the case in which only the last player is isolated. And finally, in Lemmas 3.14 and 3.15 we consider the case in which both the first and the last player are isolated. Only the proof of Lemma 3.12 is given; the proofs of Lemmas 3.13, 3.14, and 3.15 run similarly.

Lemma 3.12 *Let (P, v) be a neighbour game with $|P| \geq 3$. Assume that the matching $\mu = \{(2, 3), \dots, (p-1, p)\}$ is optimal. Define a neighbour game $(P \setminus \{1\}, \bar{v})$, by setting $\bar{v}(2, 3) := v(2, 3) - v(1, 2)$ and $\bar{v}(i, i+1) := v(i, i+1)$ for $3 \leq i \leq p-1$. Then, $n_1(v) = 0$, $n_2(v) = n_2(\bar{v}) + v(1, 2)$, and $n_i(v) = n_i(\bar{v})$ for $i \in P$, $i \geq 3$.*

Proof. Note that $\bar{v}(2, 3) \geq 0$, since $v(2, 3) \geq v(1, 2)$ by optimality of μ . So, $(P \setminus \{1\}, \bar{v})$ is indeed a neighbour game.

Let $x = n(\bar{v})$ and define $y \in \mathbb{R}^P$ by $y_1 := 0$, $y_2 := n_2(\bar{v}) + v(1, 2)$, and $y_i := n_i(\bar{v})$ for $i \in P$, $i \geq 3$. By Lemma 3.2, $y \in \text{Core}(v)$. One easily verifies that $f(\{1\}, y) = 0$, $f(\{1, 2\}, y) = f(\{2\}, x)$, and $f(\{2\}, y) \geq f(\{2\}, x) = f(\{1, 2\}, y)$. Further, it is clear that $f(S, y) = f(S, x)$ for any other singleton and pair of neighbours. Using Lemma 2.1 and Lemma 3.7 one verifies that $y = n(v)$. \square

Lemma 3.13 *Let (P, v) be a neighbour game with $|P| \geq 3$. Assume that the matching $\{(1, 2), \dots, (p-2, p-1)\}$ is optimal. Define a neighbour game $(P \setminus \{p\}, \bar{v})$, by setting $\bar{v}(p-2, p-1) := v(p-2, p-1) - v(p-1, p)$ and $\bar{v}(i-1, i) := v(i-1, i)$ for $1 \leq i \leq p-2$. Then, $n_p(v) = 0$, $n_{p-1}(v) = n_{p-1}(\bar{v}) + v(p-1, p)$, and $n_i(v) = n_i(\bar{v})$ for $i \in P$, $i \leq p-2$.*

Lemma 3.14 *Let (P, v) be a neighbour game with $|P| = 4$. Assume that the matching $\{(2, 3)\}$ is optimal. Define a neighbour game $(P \setminus \{1, 4\}, \bar{v})$, by setting $\bar{v}(2, 3) := v(2, 3) - v(1, 2) - v(3, 4)$. Then, $n_1(v) = n_4(v) = 0$, $n_2(v) = n_2(\bar{v}) + v(1, 2)$, and $n_3(v) = n_3(\bar{v}) + v(3, 4)$.*

Lemma 3.15 *Let (P, v) be a neighbour game with $|P| > 4$. Assume that the matching $\{(2, 3), \dots, (p-2, p-1)\}$ is optimal. Define a neighbour game $(P \setminus \{1, p\}, \bar{v})$, by setting $\bar{v}(2, 3) := v(2, 3) - v(1, 2)$, $\bar{v}(p-2, p-1) := v(p-2, p-1) - v(p-1, p)$, and $\bar{v}(i, i+1) := v(i, i+1)$ for $3 \leq i \leq p-3$. Then, $n_1(v) = n_p(v) = 0$, $n_2(v) = n_2(\bar{v}) + v(1, 2)$, $n_{p-1}(v) = n_{p-1}(\bar{v}) + v(p-1, p)$, and $n_i(v) = n_i(\bar{v})$ for $3 \leq i \leq p-2$.*

Now we will describe an extension of the algorithm for finding the nucleolus of an arbitrary neighbour game.

The Extended Algorithm

Let (P, v) be a neighbour game. Let μ be an optimal and minimal matching for P . Using Corollary 3.11 we break the game (P, v) up in (overlapping) subgames that are still neighbour games, but have no longer isolated players in the middle. Then, we use Lemmas 3.12, 3.13, 3.14, and 3.15 to remove the isolated players from the subgames that are not zero 2-person games. By doing this the induced matchings of μ for the obtained games may not all be minimal any longer. (See Example 3.16 for an illustration of this.) By taking an optimal and minimal matching for the games in which this occurs, we repeat the above procedure to remove the new isolated players. Eventually we have only games that satisfy either the assumptions made for the algorithm or are zero 2-person games. Once we have calculated the nucleolus of every subgame we use Lemmas 3.12, 3.13, 3.14, and 3.15 to construct the nucleolus of the game (P, v) .

Now we present an example in which the above procedure is used to calculate the nucleolus of an 11-person neighbour game.

Example 3.16 Consider the neighbour game (P, v) , where $P = \{1, \dots, 11\}$ and v is given by figure 2. The nodes depict the players and the number above an edge denotes the value of the corresponding pair of players. The numbers below the nodes are the payoffs of the nucleolus of the corresponding game. The thick edges correspond with the matched pairs in the optimal matchings. We calculate the nucleolus of (P, v) .

Using Corollary 3.11 we break the game (P, v) up in the subgames $(\{1, \dots, 6\}, v_1)$ and $(\{6, \dots, 11\}, v_5)$. Then, we use Lemma 3.15 to reduce $(\{1, \dots, 6\}, v_1)$ to $(\{2, 3, 4, 5\}, v_2)$. If we take the (unique) optimal and minimal matching $\{(4, 5)\}$ for the game $(\{2, 3, 4, 5\}, v_2)$ and apply Corollary 3.11 to $(\{2, 3, 4, 5\}, v_2)$ we get the game $(\{3, 4, 5\}, v_3)$ and the zero 2-person game $(\{2, 3\}, 0)$. Applying Lemma 3.12 to the game $(\{3, 4, 5\}, v_3)$ yields the game $(\{4, 5\}, v_4)$. For the subgame $(\{6, \dots, 11\}, v_5)$ we use Lemma 3.15 to get the game $(\{7, 8, 9, 10\}, v_6)$.

The nucleolus of the game $(\{4, 5\}, v_4)$ is $(1, 1)$. So, by Lemma 3.12 the nucleolus of $(\{3, 4, 5\}, v_3)$ is $(0, 4, 1)$. Hence, the nucleolus of $(\{2, 3, 4, 5\}, v_2)$ is $(0, 0, 4, 1)$. Then, by Lemma 3.15, the nucleolus of $(\{1, \dots, 6\}, v_1)$ is $(0, 2, 0, 4, 3, 0)$. By a similar reasoning and the use of the algorithm or Proposition 3.18(iii), the nucleolus of $(\{6, \dots, 11\}, v_5)$ is $(0, \frac{5}{3}, \frac{10}{3}, \frac{4}{3}, \frac{5}{3}, 0)$. We conclude that the nucleolus of (P, v) is $n(v) = (0, 2, 0, 4, 3, 0, \frac{5}{3}, \frac{10}{3}, \frac{4}{3}, \frac{5}{3}, 0)$. \diamond

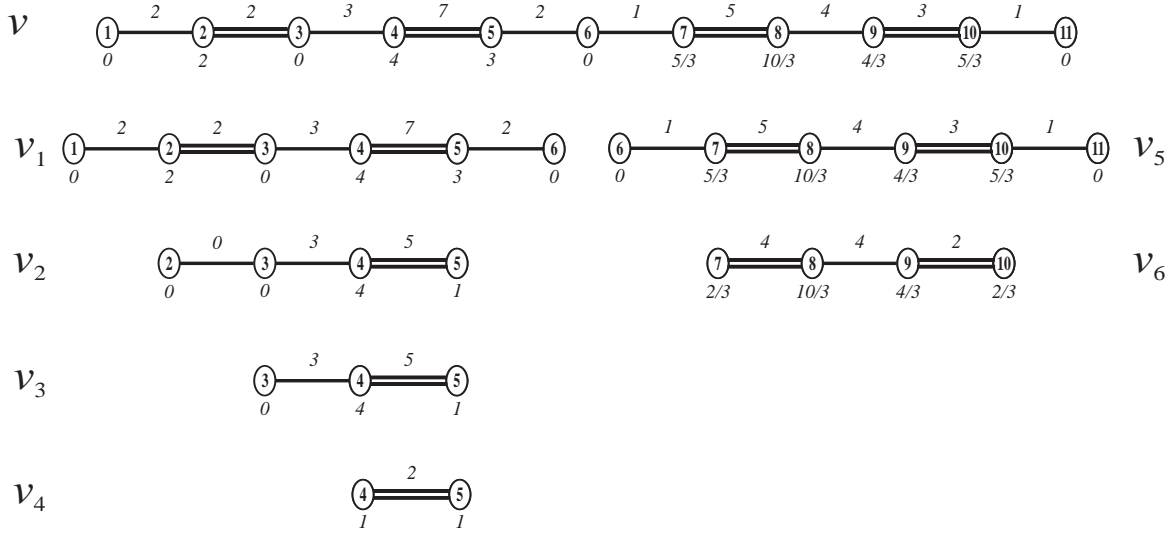


Figure 2: subsequent break-ups of the neighbour game (P, v) .

In Proposition 3.17 we show that the extended algorithm has the same computational complexity as the original algorithm.

Proposition 3.17 *The extended algorithm determines for a neighbour game (P, v) with p players the nucleolus in $\mathcal{O}(p^2)$ time.*

Proof. It takes $\mathcal{O}(p^2)$ time to find the subgames to which we apply the algorithm. Let p_1, \dots, p_k be the cardinalities of the player sets of these subgames. Note, $p_1 + \dots + p_k \leq p$. By Proposition 3.17 it takes $\mathcal{O}(p_l^2)$ time to calculate the nucleolus of the l -th subgame. Since $(p_1)^2 + \dots + (p_k)^2 \leq p^2$, it takes $\mathcal{O}(p^2)$ time to calculate the nucleoli of all subgames. Finally, note that the construction of the nucleolus of (P, v) out of the nucleoli of the subgames takes $\mathcal{O}(p^2)$ time. We conclude that we determine the nucleolus of (P, v) in $\mathcal{O}(p^2)$ time. \square

We conclude this section by providing explicit formulas for the nucleolus of neighbour games (P, v) with $|P| \leq 4$. Obviously, if $|P| = 2$ with say $v(1, 2) = a$, then $n(v) = (\frac{a}{2}, \frac{a}{2})$. Suppose $|P| = 3$. Let $a = v(1, 2)$ and $b = v(2, 3)$. We may assume, without loss of generality, that $a \geq b$. Using Lemma 2.1 one easily verifies that if $\frac{a}{2} \geq b$, then $n(v) = (\frac{a}{2}, \frac{a}{2}, 0)$, and if $\frac{a}{2} < b$, then $n(v) = (\frac{a-b}{2}, \frac{a-b}{2} + b, 0)$. The next Proposition gives an explicit formula for the nucleolus if $|P| = 4$.

Proposition 3.18 *Let (P, v) be a 4-person neighbour game, where $P = \{1, 2, 3, 4\}$ and the characteristic function v is induced by $a_{12} = a \geq 0$, $a_{23} = b \geq 0$, and $a_{34} = c \geq 0$. Assume, without loss of generality, that $a \geq c$. Then,*

(i) *if $b \in [0, \frac{c}{2})$, then $n(v) = (\frac{a}{2}, \frac{a}{2}, \frac{c}{2}, \frac{c}{2})$.*

(ii) *if $b \in [\frac{c}{2}, \frac{2a-c}{2})$, then $n(v) = (\frac{2a-2b+c}{4}, \frac{2a+2b-c}{4}, \frac{c}{2}, \frac{c}{2})$.*

(iii) *if $b \in [\frac{2a-c}{2}, a+c)$, then $n(v) = (\frac{a-b+c}{3}, \frac{2a+b-c}{3}, \frac{-a+b+2c}{3}, \frac{a-b+c}{3})$.*

(iv) *if $b \in [a+c, \infty)$, then $n(v) = (0, \frac{b+a-c}{2}, \frac{b-a+c}{2}, 0)$.*

Proof. Follows straightforwardly from Lemma 2.1. \square

References

- [1] CURIEL I., POTTERS J., RAJENDRA PRASAD V., TIJS S., AND VELTMAN B. (1994): "Sequencing and Cooperation," *Operations Research*, 42, 566-568.
- [2] DAVIS M. AND MASCHLER M. (1965): "The Kernel of a Cooperative Game," *Naval Research Logistics Quarterly*, 12, 223-259.
- [3] HAMERS H., KLIJN F., SOLYMOSI T., TIJS S., AND VILLAR J. (1999): "On the Extreme Points of the Core of Assignment Games," CentER DP 9943, Tilburg University, The Netherlands.
- [4] HUBERMAN G. (1980): "The Nucleolus and the Essential Coalitions," in Analysis and Optimization of Systems, Bensoussan A. and Lions J.L. (eds), Proceedings of the Fourth International Conference on Analysis and Springer-Verlag, Berlin/New York, 416-422.
- [5] KLIJN F., VERMEULEN D., HAMERS H., SOLYMOSI T., TIJS S., AND VILLAR J. (1999): "Neighbour Games and the Leximax Solution," CentER DP 99110, Tilburg University, The Netherlands.
- [6] KOHLBERG E. (1971): "On the Nucleolus of a Characteristic Function Game," *SIAM Journal on Applied Mathematics*, 20, 62-66.
- [7] POTTERS J. AND REIJNIERSE H. (1995): "T-Component Additive Games," *International Journal of Game Theory*, 24, 49-56.
- [8] SCHMEIDLER D. (1969): "The Nucleolus of a Characteristic Function Game," *SIAM Journal on Applied Mathematics*, 17, 1163-1170.
- [9] SHAPLEY L. AND SHUBIK M. (1972): "The Assignment Game I: The Core," *International Journal of Game Theory*, 1, 111-130.
- [10] SOLYMOSI T., AARTS H., AND DRIESSEN T. (1998): "On Computing the Nucleolus of a Balanced Connected Game," *Mathematics of Operations Research*, 23, 983-1009.
- [11] SOLYMOSI T. AND RAGHAVAN T. (1994): "An Algorithm for Finding the Nucleolus of Assignment Games," *International Journal of Game Theory*, 23, 119-143.